Context-Aware Bandits

Shuai Li
University of Insubria, Italy
shuaili.sli@gmail.com

Purushottam Kar
IIT Kanpur, India
purushot@cse.iitk.ac.in

Abstract

We propose an efficient Context-Aware clustering of Bandits (CAB) algorithm, which can capture collaborative effects. CAB can be easily deployed in a real-world recommendation system, where multi-armed bandits have been shown to perform well in particular with respect to the cold-start problem. CAB utilizes a context-aware clustering augmented by exploration-exploitation strategies. CAB dynamically clusters the users based on the content universe under consideration. We give a theoretical analysis in the standard stochastic multi-armed bandits setting. We show the efficiency of our approach on production and real-world datasets, demonstrate the scalability, and, more importantly, the significant increased prediction performance against several state-of-the-art methods.

1 Introduction

In several prominent practical applications of bandit algorithms, such as computational advertising, web-page content optimization and recommender systems, one of the main sources of information is in fact embedded in the preference relationships between the users and items served. These preference patterns that emerge from the clicks, views or purchases of items are typically exploited in Machine Learning through collaborative filtering techniques. Collaborative effects carry more information about the users preference than demographic metadata [15]. Moreover in most applications of bandit algorithms it is often impractical or impossible to use adequate user information.

In a movie recommendation system, where the catalog is relatively static and ratings for items will accumulate, one can easily deploy collaborative filtering methods such as matrix factorization or restricted Boltzmann machines. It becomes practically impossible to use the same methods in more dynamic environments as in advert or music recommendations, where there is a continuous stream of new items to be recommended along with new users. In these settings, the cold-start problem, i.e. the lack of accumulated interactions by users on items, hinders most traditional recommendation methods. These environments pose a dual challenge to recommendation methods:
1) How to present the new items to the users (or vice versa which items to present to new users) in order to optimally gather preference information on the new content (exploration)
2) How to use all the available user-item preference information gathered (exploitation).

Here one would like to ideally exploit both the content information but also more importantly the collaborative effects that in practice carry more preference information [15]. In this work we consider the dynamic recommendation setting and develop a bandit algorithm that actively learns the preference patterns among groups of users and items. Contextual bandits have been employed with success in such dynamical environments [12] but most of the focus in current approaches was on exploiting the item and user content information without taking into account collaborative effects.

When the users to serve are many and the content universe (or content popularity) changes rapidly over time, recommendation services have to show both strong adaptation in matching users' preferences and high algorithmic scalability/responsiveness so as to allow an effective online deployment. In typical scenarios like social networks, where users are engaged in technology-mediated interactions influencing each other's behavior, it is often possible to single out a few groups or *communities*

made up of users sharing similar interests. Such communities are not static over time and, more often than not, are clustered around specific content *types*, so that a given set of users can in fact host a multiplex of interdependent communities depending on specific content items, which can be changing dramatically on the fly. We call this multiplex of interdependent clusterings over users induced by the content universe as a *context-aware* clustering technique.

We consider context-aware clustering of bandits adapted to standard settings in sequential content recommendation, known as multi-armed bandits [3], for solving the associated exploration-exploitation dilemma. Note that we use the term "context" with the broader meaning as often used in the traditional bandits and not to "only" describe the features or variables under which a recommendation is served as in contextual recommendations. We work under the assumption that each content *item* determines a clustering over users made up of relatively few groups (compared to the total number of users), within which users tend to react similarly when that item gets recommended. However, the clustering over users need not be the same for different items, which is frequently observed in practice.

Our method aims to exploit collaborative effects in the bandit setting in a way akin to the way neighborhood techniques are used in batch collaborative filtering. Bandit methods represent one of the most promising approaches to the cold-start problem in recommender systems (e.g., [18]), whereby the lack of data on new users or items leads to suboptimal recommendations. We present an algorithm performing dynamic clustering for context-aware clustering and test it on production and real-world datasets. The algorithm is scalable and significantly outperforms, in terms of prediction performance, state-of-the-art bandit clustering approaches.

1.1 Related Work

One of the first works outlining stochastic multi-armed bandits for the recommendation problem is the seminal work of [12]. The first major bandit approach which sequentially clustering the users was proposed by [9]. This work led to several further developments such as [14] which utilizes the "K-Means" clustering algorithm in multi-armed bandits setting although the resulting algorithm does not perform context-aware clustering. Further advances were developed in [13] who proposed the online collaborative filtering bandits method with the interactive procedures among users and items under consideration simultaneously and the work of [10] which proposed distributed clustering of confidence ball algorithms for solving linear bandit problems in peer to peer networks.

Our proposed CAB approach distinguishes itself from previous approaches by performing context-aware clustering of users, that is able to offer impressive performance boosts in recommendation settings. Note that our model trivially subsumes the static clustering model of [9] which can be seen as assuming that all items induce the same clustering over users. Our analysis also relaxes a strict restriction imposed by [9] on the data distributions and holds for all sub-Gaussian distributions.

2 Learning Model

We design our methods, as well as analyze them in the well studied bandit clustering model that was proposed and analyzed in [9] which we introduce below. Let $\mathcal{U} = \{1, \ldots, n\}$ represent the set of n users. At each time step $t = 1, 2, \ldots$, the learner receives a user index $i_t \in \mathcal{U}$ along with a set of context vectors $C_{i_t} = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \ldots, \mathbf{x}_{t,c_t}\}$ corresponding to c_t items that can potentially be recommended to user i_t at time t. In the following, we will use the notation $\mathbf{x}_{t,k}$ to refer to both, the k^{th} item in set of items C_{i_t} presented at time t, as well as its corresponding context vector.

For the sake of simplicity, we assume that the context vectors all reside inside the unit ball in d-dimensions i.e $\mathbf{x}_{t,i} \in \mathbb{R}^d$ and $\|\mathbf{x}_{t,i}\|_2 \leq 1$ for all t and all $i \in [c_t]$. At each time step, the learner selects some $k_t \in [c_t]$, recommends the item $\mathbf{x}_{t,k_t} \in C_{i_t}$ to the user i_t, and receives a payoff a_t from the user for the recommendation. For the sake of convenience, we denote $\widehat{\mathbf{x}}_t = \mathbf{x}_{t,k_t}$.

Following the presentation of an item k_t to the user, the payoff is generated as in the linear bandit model [1, 3, 6, 7, 8, 9, 11, 12, 17, 19]. That is to say, we assume that associated with each user $i \in \mathcal{U}$, there is a (fixed but unknown) parameter $\mathbf{u}_i \in \mathbb{R}^d$, again residing inside the unit ball i.e. $\|\mathbf{u}_i\|_2 \leq 1$, which is used to generate the payoff for any item with corresponding context vector $\mathbf{x}$ as follows:

$$a_i(\mathbf{x}) = \mathbf{u}_i^\top \mathbf{x} + \epsilon_i(\mathbf{x}),$$

where $\epsilon_i(\mathbf{x})$ is a conditionally zero mean sub-Gaussian error variable, i.e $\mathbb{E}_t\left[\epsilon_i(\mathbf{x})|i_t\right] = 0$, where $\mathbb{E}_t\left[\cdot\right] := \mathbb{E}\left[\cdot|(i_1, C_{i_1}, a_1), (i_2, C_{i_2}, a_2), \ldots, (i_{t-1}, C_{i_{t-1}}, a_{t-1})\right]$, with conditional variance (defined similarly) bounded by σ^2. Note that this implies that $\mathbb{E}_t\left[a_i(\mathbf{x})|i_t\right] = \mathbf{u}_i^\top \mathbf{x}$. For the sake of clarity, we will assume that for all $i \in \mathcal{U}$ and $\mathbf{x}$, we have $a_i(\mathbf{x}) \in [-1, 1]$. We stress that our analyses do not rely upon this assumption and go through with routine modifications even if the noise values (and hence the payoffs) do not take values in a bounded range.

We will model user behavior using a family of clusterings over the user set $\mathcal{U}$. This is distinct from the work of [9] that considers a fixed and static clustering over users. We, instead, will assume that every context vector $\mathbf{x}$ induces a (potentially distinct) clustering over users. This is a much more powerful model since this allows users to agree on their opinion of certain items and disagree on others, that often holds in practice. We will call this the *context-sensitive well-separatedness* assumption. We notice that our model trivially subsumes the static clustering model of [9] which can be seen as assuming that all items induce the same clustering over the users.

Thus, we will assume that every item context $\mathbf{x}$ partitions users into $m(\mathbf{x}) \leq m$ disjoint clusters $U_1(\mathbf{x}), U_2(\mathbf{x}), \ldots, U_{m(\mathbf{x})}(\mathbf{x})$ that cover $\mathcal{U}$, i.e. $\bigcup_{i=1}^{m(\mathbf{x})} U_i(\mathbf{x}) = \mathcal{U}$ and $U_i(\mathbf{x}) \cap U_j(\mathbf{x}) = \phi$ if $i \neq j$, in such a way that users in the same cluster assign the same expected payoff when recommended that item, and users in different clusters assign expected payoffs that are bounded away from each other. More formally, let $\mathfrak{c}(\cdot, \cdot)$ denote the cluster assignment operator i.e. for any user i, and any item context $\mathbf{x}$, let $\mathfrak{c}(i, \mathbf{x}) = k \leq m(\mathbf{x})$ be the cluster for user i with respect to the item $\mathbf{x}$ i.e. $i \in U_{\mathfrak{c}(i,\mathbf{x})}(\mathbf{x})$. Then the context-sensitive well-separatedness requirement states that if $\mathfrak{c}(i, \mathbf{x}) = \mathfrak{c}(j, \mathbf{x})$ for two users i and j and some context $\mathbf{x}$, then we have $\mathbf{u}_i^\top \mathbf{x} = \mathbf{u}_j^\top \mathbf{x}$, as well as if $\mathfrak{c}(i, \mathbf{x}) \neq \mathfrak{c}(j, \mathbf{x})$ then we have $\left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| > \gamma$ for some fixed $\gamma > 0$.

Notice that neither the cluster assignment $\mathfrak{c}(\cdot, \cdot)$, nor γ are known to the learner. Also, our analyses can be easily extended to a more relaxed model where we have two distinct values for *inter-cluster* and *intra-cluster* payoff differences, i.e. have two values $\gamma_{\text{in}} < \gamma_{\text{out}}$ such that for any item $\mathbf{x}$, if $\mathfrak{c}(i, \mathbf{x}) = \mathfrak{c}(j, \mathbf{x})$, then $\left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| \leq \gamma_{\text{in}}$ and if $\mathfrak{c}(i, \mathbf{x}) \neq \mathfrak{c}(j, \mathbf{x})$ then $\left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| > \gamma_{\text{out}}$.

We note that item induced clustering of users has been studied before in the work of [13] as well as in item-based collaborative filtering approaches [16] wherein items are clustered depending on the user base that consumes them together. However, these approaches assume a finite set of items and that set needs to be explicitly maintained and manipulated, which makes these approaches expensive. Our approach is much more scalable, as well is superior to item clustering methods in practice.

Given the above problem setting, the goal of the learner is to maximize its total payoff i.e. $\sum_{t=1}^{T} a_t$ over T time steps. However, since analyzing the realized payoffs is cumbersome, we follow standard practice, and instead aim to minimize the *pseudo-regret* of the learner. Let r_t denote the *instantaneous regret* of the learner at time t, i.e. the different between the expected payoff of the item recommended and the expected payoff of the best item, i.e.

$$r_t = \left(\max_{\mathbf{x} \in C_{i_t}} \mathbf{u}_{i_t}^\top \mathbf{x}\right) - \mathbf{u}_{i_t}^\top \widehat{\mathbf{x}}_t.$$

Then the goal of the learner is to minimize $\sum_{t=1}^{T} r_t$. We note that as a special case of the above model, we can handle cases where the set of items to be recommended does not possess informative contexts. In this setting, the *non-contextual* approach [2, 4] is beneficial in making an attempt to nevertheless maximize collaborative benefits in recommendation. To implement the non-contextual approach, we simply take the set of all items $\mathcal{I}$ and apply a *one-hot* encoding by assigning the i^{th} item the context $\mathbf{e}_i$ (the i^{th} canonical basis vector with one at the i^{th} position and zeros everywhere else). It is easy to see that this encoding will require $d = |\mathcal{I}|$ dimensional contexts. Also, in this case, the expected payoff given by user i on item j is simply the j-th component of vector $\mathbf{u}_i$.

3 CAB: A Context-aware Clustering Approach to Recommendation

In this section we describe our proposed approach for context sensitive clustering of users. At a high level, our approach maintains, for each user $i \in \mathcal{U}$, an estimate $\mathbf{w}_i$ of the true user model vector $\mathbf{u}_i$, as that is standard in recommendation systems. This is done by operating a linear bandit algorithm [1, 5, 6, 9] at every user. At every time step t, we receive a user i_t to serve, and a set of item contexts

Algorithm 1 CAB: Context-aware Clustering of Bandits

Input: Exploration parameter $\alpha > 0$

$\mathbf{b}_{i,0} = \mathbf{0} \in \mathbb{R}^d$ and $M_{i,0} = I \in \mathbb{R}^{d \times d}$, $i = 1, \ldots n$.
for $t = 1, 2, \ldots, T$ **do**
  for every user $i = 1, \ldots, n$ **do**
    Set $\mathbf{w}_{i,t-1} = M_{i,t-1}^{-1} \mathbf{b}_{i,t-1}$
    Set $\text{CB}_{i,t-1}(\mathbf{x}) = \alpha \sqrt{\mathbf{x}^\top M_{i,t-1}^{-1} \mathbf{x} \, \log(t+1)}$
  end for
  Receive user $i_t \in \mathcal{U}$, and items $C_{i_t} = \{\mathbf{x}_{t,1}, \ldots, \mathbf{x}_{t,c_t}\}$ to be recommended
  `//Prepare item specific neighborhoods and payoff estimates`
  for every item context $k = 1, \ldots, c_t$ **do**
    Compute the neighborhood for i_t for this item
    $$N_{t,k} =: N_{i_t,t}(\mathbf{x}_{t,k}) = \left\{ j \in \mathcal{U} \; : \; |\mathbf{w}_{i_t,t-1}^\top \mathbf{x}_{t,k} - \mathbf{w}_{j,t-1}^\top \mathbf{x}_{t,k}| \leq \text{CB}_{i_t,t-1}(\mathbf{x}_{t,k}) + \text{CB}_{j,t-1}(\mathbf{x}_{t,k}) \right\}$$
    Compute the neighborhood level aggregate payoff and confidence bound using CAB1 or CAB2
    $$(\overline{a}_{t,k}, \overline{c}_{t,k}) = \text{GET-AGGREGATE}(N_{t,k}, \mathbf{x}_{t,k})$$
  end for
  Set $k_t = \arg\max_{k=1,\ldots,c_t} \{\overline{a}_{t,k} + \overline{c}_{t,k}\}$,
  Recommend the item $\widehat{\mathbf{x}}_t := \mathbf{x}_{t,k_t}$ to the user i_t and observe payoff $a_t \in \mathbb{R}$
  `//Update Local Linear Bandit Estimates`
  $M_{i_t,t} = M_{i_t,t-1} + \widehat{\mathbf{x}}_t \widehat{\mathbf{x}}_t^\top$
  $\mathbf{b}_{i_t,t} = \mathbf{b}_{i_t,t-1} + a_t \widehat{\mathbf{x}}_t$
  $M_{i,t} = M_{i,t-1}$, $\mathbf{b}_{i,t} = \mathbf{b}_{i,t-1}$ for all $i \neq i_t$
end for

Algorithm 2 CAB1: Average Aggregate

Input: Cluster $N_{t,k}$, item context $\mathbf{x}_{t,k}$

`//Compute aggregate model estimate`
$\overline{\mathbf{w}}_{t,k} = \frac{1}{|N_{t,k}|} \sum_{j \in N_{t,k}} \mathbf{w}_{j,t-1}$
`//Compute aggregate payoff estimate`
$\overline{a}_{t,k} = \overline{\mathbf{w}}_{t,k}^\top \mathbf{x}_{t,k}$
`//Compute aggregate confidence bound`
$\overline{c}_{t,k} = \frac{1}{|N_{t,k}|} \sum_{j \in N_{t,k}} \text{CB}_{j,t-1}(\mathbf{x}_{t,k})$
return $(\overline{a}_{t,k}, \overline{c}_{t,k})$

Algorithm 3 CAB2: Corrective Aggregate

Input: Cluster $N_{t,k}$, item context $\mathbf{x}_{t,k}$

`//Unravel neighbor histories`
$\overline{M}_{t,k} = I + \sum_{i \in N_{t,k}} (M_{i,t-1} - I)$
$\overline{\mathbf{b}}_{t,k} = \sum_{i \in N_{t,k}} \mathbf{b}_{i,t-1}$
`//Compute aggregate model estimate`
$\overline{\mathbf{w}}_{t,k} = \overline{M}_{t,k}^{-1} \overline{\mathbf{b}}_{t,k}$
`//Compute aggregate payoff estimate`
$\overline{a}_{t,k} = \overline{\mathbf{w}}_{t,k}^\top \mathbf{x}_{t,k}$
`//Compute aggregate confidence bound`
$\overline{c}_{t,k} = \alpha \sqrt{\mathbf{x}_{t,k}^\top \overline{M}_{t,k}^{-1} \mathbf{x}_{t,k} \, \log(t+1)}$
return $(\overline{a}_{t,k}, \overline{c}_{t,k})$

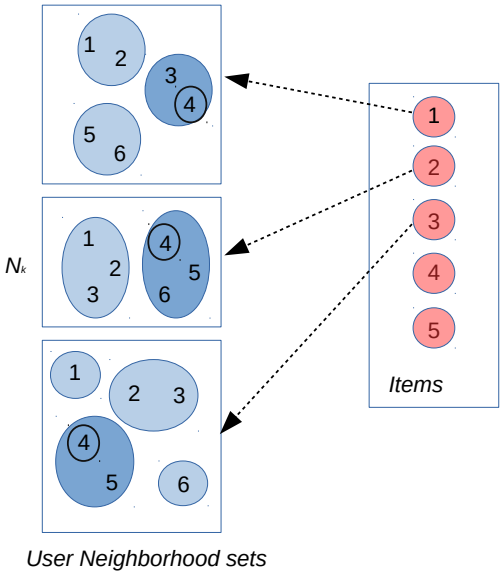

Figure 1: The CONTEXT-AWARE BANDIT algorithm and its two variants.

$C_{i_t} = \{\mathbf{x}_{t,1}, \mathbf{x}_{t,2}, \ldots, \mathbf{x}_{t,c_t}\}$ to choose from. We use these contexts to perform a context sensitive clustering of users.

To perform this clustering, we look at every item context $\mathbf{x}_{t,k}$ in turn, and find all users who are expected to give similar payoffs to item $\mathbf{x}_{t,k}$ as user i_t would. To do so we make use of the estimates $\mathbf{w}_i$ of the true user model vector that we have maintained so far to get estimates of the expected payoffs. All users whose estimated payoff match those of user i_t upto confidence intervals are

collected in a *neighborhood* $N_{i_t,t}(\mathbf{x}_{t,k}) =: N_{t,k}$. Such neighborhoods are identified for every item context $\mathbf{x}_{t,k}$. The exact procedure for this is outlined in Algorithm 1.

Having constructed these neighborhoods, we next exploit the collaborative effects of having similar minded users in each neighborhood. Thus, for each item, we look at its corresponding neighborhood, and aggregate user history for users in that neighborhood to get upper-confidence style estimates of the expected payoff for that item. We propose two methods to perform this aggregation: the first algorithm, called CAB1 (see Algorithm 2), performs an intuitive, and inexpensive step of simply averaging of the model estimates $\mathbf{w}_j$ of users j in the neighborhood and using that to obtain the aggregate neighborhood level estimate of the payoff given to a certain item. A similar step is done to obtain an upper confidence bound for this payoff estimate as well.

The second approach, called CAB2 (see Algorithm 3), performs a more detailed analysis of the users who form a part of this neighborhood. The algorithm unravels the user histories for all users in that neighborhood and constructs aggregate neighborhood level payoff estimates, as well as confidence bounds afresh from these aggregated user histories. Arguably this requires more processing but, as we shall see, this algorithm offers more accurate recommendations in practice as well.

Finally the algorithm uses these payoff estimates and upper confidence bounds to execute a UCB-style exploration-exploitation step to choose the item to be recommended. The payoff received then is use to update the model and confidence bound estimates of the user i_t being served at that stage according to the linear bandit framework. Notice that the update is only performed at user i_t, although this update will affect the calculation of neighborhood sets and compound vectors for other users in later rounds. We also notice that the two versions of the CAB algorithm are minimal in the amount of aggregation they perform to obtain collaborative estimates of payoffs, as well as are very intuitive. In the following, we prove regret bounds for both versions of CAB in the standard stochastic setting.

4 Regret Analysis

To present our regret analysis, we make some additional assumptions about the learning model. We note that these assumptions are standard in clustering bandit literature [9, 13]. At each time t, the user i_t to be served is chosen uniformly randomly from the set of users $\mathcal{U}$. We stress that this assumption of uniformity over users is simply for the sake of convenience and not critical to our analysis. Our proofs are easily modified to handle any non-uniform distribution that has support over the entire user set $\mathcal{U}$ i.e. which does not starve any user from being served.

Once the user has been selected, the number of items to be put up for recommendation c_t is chosen arbitrarily given the past users served, context sets, payoffs, and the current user identity. Next the context set C_{i_t} is generated, conditioned on past history and i_t, by sampling c_t vectors from a distribution $\mathcal{D}$ over $\mathbb{R}^d$. We assume that $\mathcal{D}$ possesses a full rank covariance matrix i.e. if $X \sim \mathcal{D}$ then $\mathbb{E}\left[XX^\top\right] \succeq \lambda I$ for some $\lambda > 0$. Also, for any fixed vector $\mathbf{z} \in \mathbb{R}^d$, let the random variable $(\mathbf{z}^\top X)^2$ be sub-Gaussian, with its variance parameter at most ν^2, i.e. $\mathbb{E}_t\left[\exp(\alpha(\mathbf{z}^\top X)^2)|(i_t, c_t)\right] \leq \exp(\alpha^2\nu^2/2)$ for any $\alpha > 0$. We note that these are standard assumptions.

At any time step t we define the following quantities

1. (Best item in hindsight) Let $k_t^* = \arg\max_{k\in[c_t]} \mathbf{u}_{i_t}^\top \mathbf{x}_{t,k}$ denote the identity of the best item and let $\mathbf{x}_t^* = \mathbf{x}_{t,k_t^*}$ denote its corresponding context vector.
2. (Recommended item) Using the notation used in Algorithm 1, we let k_t denote the identity of the item recommended to the user at time t (Algorithm 1 line 12) and let $\widehat{\mathbf{x}}_t$ denote its context vector.
3. (True neighborhoods) We let $N_t^* = \mathfrak{c}(i_t, \mathbf{x}_t^*)$ denote the true neighborhood for the user i_t with respect to the best item $\mathbf{x}_t^*$ and let $\widetilde{N}_t = \mathfrak{c}(i_t, \widehat{\mathbf{x}}_t)$ denote the true neighborhood of the user i_t with respect to the recommended item $\widehat{\mathbf{x}}_t$.
4. (Estimated neighhborhoods) We let $\widehat{N}_t^* := N_{t,k_t^*}$ denote the estimated neighborhood of the user i_t with respect to the best item $\mathbf{x}_t^*$ and let $\widehat{N}_t := N_{t,k_t}$ denote the estimated neighborhood of the user i_t with respect to the recommended item $\widehat{\mathbf{x}}_t$ (Algorithm 1 line 9).

For the sake of convenience, while proving our bounds, we will assume that the algorithms are executed with the confidence bounds $\text{CB}_{i,t}(\mathbf{x})$ with their "theoretical" counterparts. We will assume that Algorithm 1 and Algorithm 2 (the CAB1 update), use the following confidence bound:

$$\text{TCB}_{i,t}(\mathbf{x}) = \sqrt{\mathbf{x}^\top M_{i,t}^{-1}\mathbf{x}}\left(\sigma\sqrt{2\log\frac{2\left|M_{i,t}\right|}{\delta}}+1\right).$$

For Algorithm 3 (the CAB2 update), we will use the following confidence bound:

$$\overline{\text{TCB}}_{t,k}(\mathbf{x}) = \sqrt{\mathbf{x}^\top \overline{M}_{t,k}^{-1}\mathbf{x}}\left(\sigma\sqrt{2\log\frac{2\left|\overline{M}_{t,k}\right|}{\delta}}+n\right),$$

where $\overline{M}_{t,k} = I + \sum_{j\in N_{t,k}}(M_{j,t-1} - I)$, as defined in Algorithm 3.

We note that this assumption is not crucial to our analyses and there to make the exposition simpler. The algorithms are not executed with these theoretical confidence bounds since they are more expensive to evaluate as they involve the determinants of $d \times d$ matrices. Given this, we prove the following basic lemma which is at the core of our regret analyses. This lemma dictates how quickly are the CAB variants able to correctly identify the item-sensitive neighborhoods for different users.

Lemma 1. *With probability at least $1-\delta$, uniformly over all pairs of users $i, j \in \mathcal{U}$ and all $t = 1, 2, \ldots$, the following results hold*

1. *If $\left|\mathbf{u}_i^\top\mathbf{x} - \mathbf{u}_j^\top\mathbf{x}\right| > \gamma$ and $\text{TCB}_{i,t}(\mathbf{x}) + \text{TCB}_{j,t}(\mathbf{x}) \leq \gamma/2$, then $\left|\mathbf{w}_{i,t}^\top\mathbf{x} - \mathbf{w}_{j,t}^\top\mathbf{x}\right| > \text{TCB}_{i,t}(\mathbf{x}) + \text{TCB}_{j,t}(\mathbf{x})$*
2. *If $\left|\mathbf{w}_{i,t}^\top\mathbf{x} - \mathbf{w}_{j,t}^\top\mathbf{x}\right| > \text{TCB}_{i,t}(\mathbf{x}) + \text{TCB}_{j,t}(\mathbf{x})$, then $\left|\mathbf{u}_i^\top\mathbf{x} - \mathbf{u}_j^\top\mathbf{x}\right| > \gamma$.*

Recall that γ is the parameter in the context-sensitive well-separatedness requirement which implies thta if $\mathfrak{c}(i,\mathbf{x}) = \mathfrak{c}(j,\mathbf{x})$ for two users i and j and some context $\mathbf{x}$, then we have $\mathbf{u}_i^\top\mathbf{x} = \mathbf{u}_j^\top\mathbf{x}$, as well as if $\mathfrak{c}(i,\mathbf{x}) \neq \mathfrak{c}(j,\mathbf{x})$ then we have $\left|\mathbf{u}_i^\top\mathbf{x} - \mathbf{u}_j^\top\mathbf{x}\right| > \gamma$.

We note that part 2 of the Lemma assures us that for no item $\mathbf{x}_{t,k} \in C_{i_t}$, will Algorithm 1 ever place two users in different clusters if they actually agree on the payoff for that item. Thus, the algorithm will never unnecessarily split users who agree on a particular item, into two or more clusters. However, the algorithm may produce impure clusters, which contain users who do not agree regarding the item. Part 1 of the lemma assures us that even this will not happen, once the confidence bounds $\text{TCB}_{i,t}(\mathbf{x})$ are appropriately low. Thus, Lemma 1 immediately gives us the following corollary:

Corollary 2. *At any time step t and any item $\mathbf{x}_{t,k} \in C_{i_t}$, if we have $\max_{i\in\mathcal{U}} \text{TCB}_{i,t}(\mathbf{x}_{t,k}) \leq \gamma/4$, then we have $N_{t,k} = U_{\mathfrak{c}(i_t,\mathbf{x}_{t,k})}(\mathbf{x}_{t,k})$, i.e. step 9 in Algorithm 1 correctly identifies the neighborhood for the user i_t with respect to the item $\mathbf{x}_{t,k}$.*

This is a very useful corollary since this guarantees perfect identification of the neighborhood with respect to a given item. In the next step, we establish a bound on the terms $\text{TCB}_{i,t}(\mathbf{x})$ that is uniform over all contexts $\mathbf{x}$, all users $i \in \mathcal{U}$, and all time steps t. A similar bound was proven in [9, Lemma 2]. However, that result assumes that the values of λ and ν^2, respectively the smallest eigenvalue of the covariance matrix associated with the distribution $\mathcal{D}$ and its conditional sub-Gaussian variance parameter, satisfy the following relation at each time step

$$\nu^2 \leq \frac{\lambda^2}{8\log(4c_t)}$$

Note that this imposes the restriction $c_t \leq \exp(\lambda^2/8\nu^2)$, i.e restricts the number of items their method can handle at any time step, to a small constant. The above relation may not even hold for general distributions. Our result, proved below, makes no such assumptions about λ and ν^2 and allows them to take positive values freely. In the following, for any user $i \in \mathcal{U}$ we will use $T_{i,t}$ to denote the number of times user i has been served till time t, i.e. $T_{i,t} = \sum_{\tau=1}^{t}\mathbb{I}\{i_\tau = i\}$. Also, for any neighborhood $N \subseteq \mathcal{U}$ of users, we will use $T_{N,t} = \sum_{j\in N} T_{j,t}$ to denote the total number of times users in the neighborhood N have been served in the past.

Lemma 3. *Assume that users and contexts are generated according to the model specified in Section 2. Then, with probability $1-\delta$, the following holds uniformly over all time steps $t = 1, 2, \ldots$, all users $i \in \mathcal{U}$ and all contexts $\mathbf{x} \in \mathbb{R}^d$ of less than unit norm,*

$$\textsc{tcb}_{i,t}(\mathbf{x}) \leq M_{d,\lambda,\mu,\delta}(i,t) := \frac{3\sigma\sqrt{d\log t + \log(1/\delta)}}{\sqrt{1 + B_{\lambda,\nu}(T_{i,t}, \delta/2nd)}},$$

where $B_{\lambda,\nu}(T,\delta) = (C(\lambda,\nu)\cdot T - 8(\log(T/\delta) + \sqrt{T\log(T/\delta)}))_+$ and $C(\lambda,\nu)$ is a strictly positive constant that depends on λ, the smallest eigenvalue of the covariance matrix associated with the distribution $\mathcal{D}$ and ν^2, its conditional sub-Gaussian variance parameter.

This bound has two desirable properties that are missing from the result of [9]. Firstly, we note that for our regret bounds to go through, it is sufficient that $C(\lambda,\nu)$ take any strictly positive value. We notice that Lemma 3 is able to assure this for *any* values of λ and ν^2. Secondly, the result also allows the number of items being considered at any time step t to be an arbitrarily large constant.

A similar bound holds for the theoretical bounds for the CAB2 algorithm as well and can be proved in a similar manner by taking a union bound over all 2^n possible neighborhoods.

Lemma 4. *Assume that users and contexts are generated according to the model specified in Section 2. Then, with probability $1-\delta$, the following holds uniformly over all time steps $t = 1, 2, \ldots$ and all context vectors $\mathbf{x}_{t,k} \in C_{i_t}$ at time t,*

$$\overline{\textsc{tcb}}_{t,k}(\mathbf{x}_{t,k}) \leq \overline{M}_{d,\lambda,\mu,\delta}(t,k) := \frac{3\sigma\sqrt{d\log t + \log(1/\delta)} + n}{\sqrt{1 + B_{\lambda,\nu}(T_{N_{t,k},t}, \delta/2^n d)}},$$

where $B_{\lambda,\nu}(T,\delta) = (C(\lambda,\nu)\cdot T - 8(\log(T/\delta) + \sqrt{T\log(T/\delta)}))_+$ and $C(\lambda,\nu)$ is a strictly positive constant that depends on λ, the smallest eigenvalue of the covariance matrix associated with the distribution $\mathcal{D}$ and ν^2, its conditional sub-Gaussian variance parameter.

Lemma 3 is a very powerful result since it assures us that after $T_{i,t}$ has grown sufficiently large, i.e. the user i has been served sufficiently enough, we will have $\textsc{tcb}_{i,t}(\mathbf{x})$ for *every* context vector $\mathbf{x}$. This, combined with Corollary 2 will then ensure that the neighborhood for user i_t with respect to the item $\mathbf{x}$ will get identified exactly. With this, we are ready to establish a regret bound for Algorithm1 executed with the CAB1 aggregation step.

4.1 A Regret Bound for the Average Aggregation Method CAB1

We recall that the aim of this result is to establish an upper bound on the quantity

$$\sum_{t=1}^{T} r_t = \sum_{t=1}^{T} \mathbf{u}_{i_t}^\top (\mathbf{x}_t^* - \widehat{\mathbf{x}}_t).$$

In this section, we establish the following regret bound for the CAB1 algorithm.

Theorem 5. *Algorithm 1, when executed with the average aggregation CAB1 step, guarantees with probability at least $1-\delta$, the following pseudo-regret bound (neglecting universal constants for the sake of clarity)*

$$\sum_{t=1}^{T} r_t \leq \sigma\sqrt{d\log(T/\delta)} \cdot \left(n\left(\frac{1}{C(\lambda,\nu)^3} \log^2 \frac{1}{\delta} + \log\frac{nT}{\delta} \right) + \sqrt{\frac{nT}{C(\lambda,\nu)}} \right) + \frac{n\sigma^2 d\log(T/\delta)}{C(\lambda,\nu)\cdot\gamma^2}$$

If we look at only the dominating term in the above expression, we get

$$\sum_{t=1}^{T} r_t \leq \sigma\sqrt{\frac{ndT\log\frac{T}{\delta}}{C(\lambda,\nu)}} + \widetilde{\mathcal{O}}(1).$$

To prove this bound, we first recall that we had assumed for simplicity, that the model vectors $\mathbf{u}_i$ and context vectors $\mathbf{x}_{t,k} \in C_{i_t}$ all have less than unit Euclidean norm. This means that $r_t \leq 2$.

$$\sum_{t=1}^{T} r_t = \sum_{t=1}^{T} r_t \cdot \mathbb{I}\left\{ \max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t) \leq \gamma/4 \right\} + \sum_{t=1}^{T} r_t \cdot \mathbb{I}\left\{ \max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t) > \gamma/4 \right\}$$

$$\leq \underbrace{\sum_{t=1}^{T} r_t \cdot \mathbb{I}\left\{\max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t) \leq \gamma/4\right\}}_{(A)} + 2 \cdot \underbrace{\sum_{t=1}^{T} \mathbb{I}\left\{\max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t) > \gamma/4\right\}}_{(B)}.$$

We bound the two quantities (A) and (B) separately below. The proof for (B) is simpler, and hinges on the fact that users to be served at each time step are chosen uniformly and randomly. Thus, after enough time steps, every user will be served sufficiently many times i.e. $T_{i,t}$ will be sufficiently large. This, combined with the definition of $M_{d,\lambda,\mu,\delta}(i,t)$ will establish the bound on (B). We recall here that the assumption that users are chosen uniformly at random is not critical to our proof and the bound on (B) can be established for any distribution over the users which has support over $\mathcal{U}$.

Lemma 6. *With probability at least $1-\delta$, we have, for both the* CAB*1 and the* CAB*2 updates*

$$(B) \leq 2n\left(\max\left\{\frac{32^3}{C(\lambda,\nu)^3}\log^2\frac{1}{\delta}, \frac{288\sigma^2\left(d\log T + \log(1/\delta)\right)}{C(\lambda,\nu)\cdot\gamma^2}\right\} + 2\log\frac{nT}{\delta}\right).$$

The bound on the quantity (A) is more involved and makes use of the fact that since terms in the quantity are conditioned on $M_{d,\lambda,\mu,\delta}(i,t)$ being sufficiently small, using Corollary 2, we can assume that the neighborhoods for the user i_t were identified correctly for all contexts $\mathbf{x} \in C_{i_t}$. This, combined with standard UCB-style manipulations, establish a bound on (A).

Lemma 7. *With probability at least $1-\delta$, we have, for the* CAB*1 update*

$$(A) \leq 3\sigma\sqrt{d\log T + \log(1/\delta)}\cdot\left(2n\left(\frac{32^3}{C(\lambda,\nu)^3}\log^2\frac{1}{\delta} + 2\log\frac{nT}{\delta}\right) + 4n\log\frac{nT}{\delta} + \frac{8n}{C(\lambda,\nu)} + 4\sqrt{\frac{nT}{C(\lambda,\nu)}}\right)$$

Combining the two bounds establishes Theorem 5.

4.2 A Regret Bound for the Corrective Aggregation Method CAB2

We now move on to establish a similar regret bound for the CAB2 update. As before, To make the exposition simple, we will require a slight modification to the "theoretical" confidence bounds that we have been working with, as we mentioned earlier. Recall that for the CAB2 analysis, we will assume that the following confidence bounds are used:

$$\overline{\text{TCB}}_{t,k}(\mathbf{x}) = \sqrt{\mathbf{x}^\top \overline{M}_{t,k}^{-1}\mathbf{x}}\left(\sigma\sqrt{2\log\frac{2\left|\overline{M}_{t,k}\right|}{\delta}} + n\right),$$

where $\overline{M}_{t,k} = I + \sum_{j\in N_{t,k}}(M_{j,t-1} - I)$, as defined in Algorithm 3. We will seek to establish the following regret bound.

Theorem 8. *Algorithm 1, when executed with the average aggregation* CAB*1 step on users with uniform neighborhoods with respect to all contexts, guarantees with probability at least $1-\delta$, the following pseudo-regret bound (neglecting universal constants for the sake of clarity)*

$$\sum_{t=1}^{T} r_t \leq \sigma(\sqrt{d\log(T/\delta)}+n)\cdot\left(n\left(\frac{n^3}{C(\lambda,\nu)^3}\log^2\frac{1}{\delta} + n\log\frac{T}{\delta}\right) + \sqrt{\frac{nT}{C(\lambda,\nu)}}\right) + \frac{n\sigma^2 d\log(T/\delta)}{C(\lambda,\nu)\cdot\gamma^2}$$

If we look at only the dominating term in the above expression, we get

$$\sum_{t=1}^{T} r_t \leq \sigma\sqrt{\frac{ndT\log\frac{T}{\delta}}{C(\lambda,\nu)}} + \widetilde{\mathcal{O}}(1),$$

which is similar as the bound we arrived at for CAB1. However, the above bound for CAB2 is worse in the non-dominating terms since it has an n^4 dependence in one of the constant terms (i.e. terms that do not depend on T). We note that this can be greatly improved by doing a slightly more

careful analysis as follows: in the above bound, the worse dependence is introduced by a union bound over all possible neighborhoods that items can induce over the set of users.

Currently, the analysis takes the possible number of neighborhoods as 2^n, which is a very wasteful bound since frequently (see [13] for instance) items taken together induce much less than 2^n clusterings over the users. If $\aleph \ll 2^n$ is the number of clusterings all possible items induce over the users, then the above bound in Theorem 8 can be improved by replacing n with $\log \aleph \ll n$. We postpone these finer bounds to a later version of the paper.

The proof of this result will proceed in a manner similar to that for the CAB1 update, by decomposing the regret into the two terms (A) and (B). The bound on the term (B) will continue to hold as before and will be reused here. However, the bound on (A) has to be established separately.

Bounding (A) will require novel results that involve a fine grained analysis of the interaction between the model vectors and the contexts, as well as establishing a refined version of the *Confidence Ellipsoid* result of [1, Theorem 2], which we do below. We note that existing results fail to apply in our setting since our algorithm clusters users that may differ from each other significantly otherwise, but agree just on a given item. Existing results on linear bandits hold only for each individual user, not when data from multiple users, with significantly different profiles, are combined together.

To present the bound, we introduce the notion of *discrepancy* in a neighborhood. At any time step t, and any context $\mathbf{x}_{t,k} \in C_{i_t}$, let $N = \mathfrak{c}(i_t, \mathbf{x}_{t,k})$ denote the true neighborhood of the user i_t with respect to the item $\mathbf{x}_{t,k}$. Then this neighborhood is said to have discrepancy $\kappa_{i_t,\mathbf{x}_{t,k}}$, if $\max_{i,j \in N} \left\| \mathbf{u}_i - \mathbf{u}_j \right\|_2 \leq \kappa_{i_t,\mathbf{x}_{t,k}}$. The discrepancy of a neighborhood measures how distinct are the users who have formed a part of this neighborhood.

We clarify that the analysis of the CAB1 approach (as outlined in the previous section) does, in no way, require notions of discrepancy or any bounds thereon. This notion is needed solely for the analysis of the CAB2 approach.

Theorem 9. *With probability at least $1-\delta$, uniformly over time steps $t > 0$, users $i \in \mathcal{U}$, and context vectors $\mathbf{x}_{t,k} \in C_{i_t}$, we have*

$$\left| \left(\overline{\mathbf{w}}_{t,k} - \frac{1}{|N|} \sum_{j \in N} \mathbf{u}_j \right)^\top \mathbf{x}_{t,k} \right| \leq \overline{\mathrm{TCB}}_{t,k}(\mathbf{x}_{t,k}) + \mathcal{O}\left(\kappa_{i_t,\mathbf{x}_{t,k}} \right),$$

where $N = \mathfrak{c}(i_t, \mathbf{x})$ denotes the true neighborhood of the user i_t with respect to an item $\mathbf{x}_{t,k} \in C_{i_t}$ and $\kappa_{i_t,\mathbf{x}_{t,k}}$ denotes the discrepancy in the neighborhood of the user i_t with respect to the item $\mathbf{x}_{t,k}$.

We note that this result is much more precise than similar results in previous literature [1, 9] which go ahead and establish bounds on $\left\| \mathbf{w}_{j,t} - \mathbf{u}_j \right\|_2$ [1] or $\left\| \overline{\mathbf{w}}_{t,k} - \frac{1}{|N|} \sum_{j \in N} \mathbf{u}_j \right\|_2$ [9]. No such result is possible here since the users that have come together in the neighborhood N may wildly disagree on every item other than $\mathbf{x}_{t,k}$, the item in question. Thus, Theorem 9 seems to be the only plausible bound without making further assumptions about the learning model. We note that this gives us a very convenient result that tells us that the aggregate model vector calculated by the CAB2 algorithm actually gives very faithful estimates of the payoffs the users in that neighborhood give to that item.

Using this result we now prove a bound on the term (A). We will prove the result for the case of uniform neighborhoods i.e. settings where we have context whose neighborhoods have no discrepancy. We will call such neighborhoods *uniform*. This condition is satisfied, for example in the bandit clustering setting where users are clustered into groups where users in a group share the same model vector. In this case there is no discrepancy in the neighborhoods. We note that we can also present a regret bound in the general case. However, in that case, our regret bound will have an additional additive factor of the order of $\kappa_{\max} \cdot T$ where $\kappa_{\max} = \max_t \max_{\mathbf{x} \in C_{i_t}} \kappa_{i_t,\mathbf{x}}$.

Lemma 10. *With probability at least $1-\delta$, for the* CAB2 *update with uniform neighborhoods,*

$$(A) \leq 6(\sigma\sqrt{d \log T + \log(1/\delta)} + n) \cdot \left(2n \left(\frac{(32n)^3}{C(\lambda,\nu)^3} \log^2 \frac{1}{\delta} + 2n \log \frac{T}{\delta} \right) + 4n^2 \log \frac{T}{\delta} + \frac{8n}{C(\lambda,\nu)} + 4\sqrt{\frac{nT}{C(\lambda,\nu)}} \right)$$

Together with Lemma 6, this establishes Theorem 8.

5 Experiments

We tested our algorithms on production and real-world benchmark datasets and compared them to a number of state-of-the-art bandit baselines. In all cases, we used the one-hot encoding, and we implemented the same experimental setting as in [5, 9, 10, 13])[1].

5.1 Dataset Description

Tuenti. This production line data was prepared by Tuenti.com, a Spanish social network website. It contains ad impressions viewed by user along with clicks on ads. The number of available ads, users and records/rounds/timesteps, turned out to be $d = 105$, $n = 14,612$, and $T = 1,000,000$. Since the available payoffs are those associated with the items served by system, we discard on the fly all records where system's recommendation did not coincide with the algorithms' recommendations. We simulated random choices by constructing the available item sets C_{i_t} as follows. At each round t, we retained the ad served to the current user i_t and the associated payoff value a_t (1 = "clicked", 0 = "not clicked"). We then created C_{i_t} by including the served ad along with 14 extra items (hence $c_t = 15$ $\forall t$) drawn uniformly at random in such a way that, for any item $\mathbf{e}_j \in \mathcal{I}$, if $\mathbf{e}_j$ occurs in some set C_{i_t}, this item will be the one served by system only $1/15$ of the times, and notice that this random selection was done independent of the available payoff a_t.

KDD Cup. This dataset was released for the KDD Cup 2012 Online Advertising Competition[2] where the instances derived from the session logs of a search engine (soso.com of Tencent Inc.). One search session refers to an interaction between a user and the search engine. It contains the following data: the user, the query issued by the user, some ads returned by the search engine and thus impressed (displayed) to the user, and the ads that were clicked by the user (zero otherwise). Each session was divided into multiple instances, where each instance describes an impressed ad under a certain depth and position. Instances were aggregated with the same user id, ad id, and query. We took the chronological order among all the instances, and we dropped the first 20 instances in order to initialize the arms/ads pool with the length of recommendation list $c_t = 20$, the resulting datasets ended up with $T = 100,000$, $n = 10,333$ distinct users, and $d = 6,780$ distinct ads.

Avazu. This dataset was circulated by the Avazu Click-Through Rate Prediction Challenge on Kaggle platform[3] where the click-through data ordered chronologically, and non-clicks and clicks are subsampled according to different strategies. We conducted the similar procedure as above and it turns out to be $n = 48,723$, $c_t = 20$, $d = 5,099$, and $T = 1,138,405$.

LastFM. This is a dataset built from the streaming Last.fm website logs, it includes the listening sessions of about $1,000$ users, each one representing a Last.fm user listening to a song. The part of the original dataset we used for this experiment is a list of tuples defining time, user, and listened song. The dataset was not created to be used for experiments with multi-armed bandits, we thus had to enrich the dataset. Each list was made up of the song that the current user listened to (with payoff 1) along with a set of candidate songs selected uniformly at random from the available songs (with payoff 0). The experiments carried out over $T = 100,000$, resulting in recommendation list's length $c_t = 20$ with $d = 4,698$ distinct songs.

5.2 Baseline Measures

We used the first 20% data of each dataset for training and the rest for testing, all experimental results were averaged over 5 runs. Tunable parameters were picked via standard grid search on the training sets and then we stick to those best parameters for reporting the performance behavior on the test sets. We compared our algorithms to a number of state-of-the-art bandit methods:

- CLUB [9] is the state-of-the-art clustering of bandits in centralised environment, it sequentially clusters the users based on their confidence ellipsoid balls;
- DCCB [10] is the state-of-the-art clustering of bandits in distributed computing setting;
- COFIBA [13] is the state-of-the-art interactive bandits method conducts the online clustering for both users and items under collaborative filtering scenario;

[1] In practice we only take a subset of users at each time step to compute neighbors.

[2] http://www.kddcup2012.org/c/kddcup2012-track2

[3] https://www.kaggle.com/c/avazu-ctr-prediction

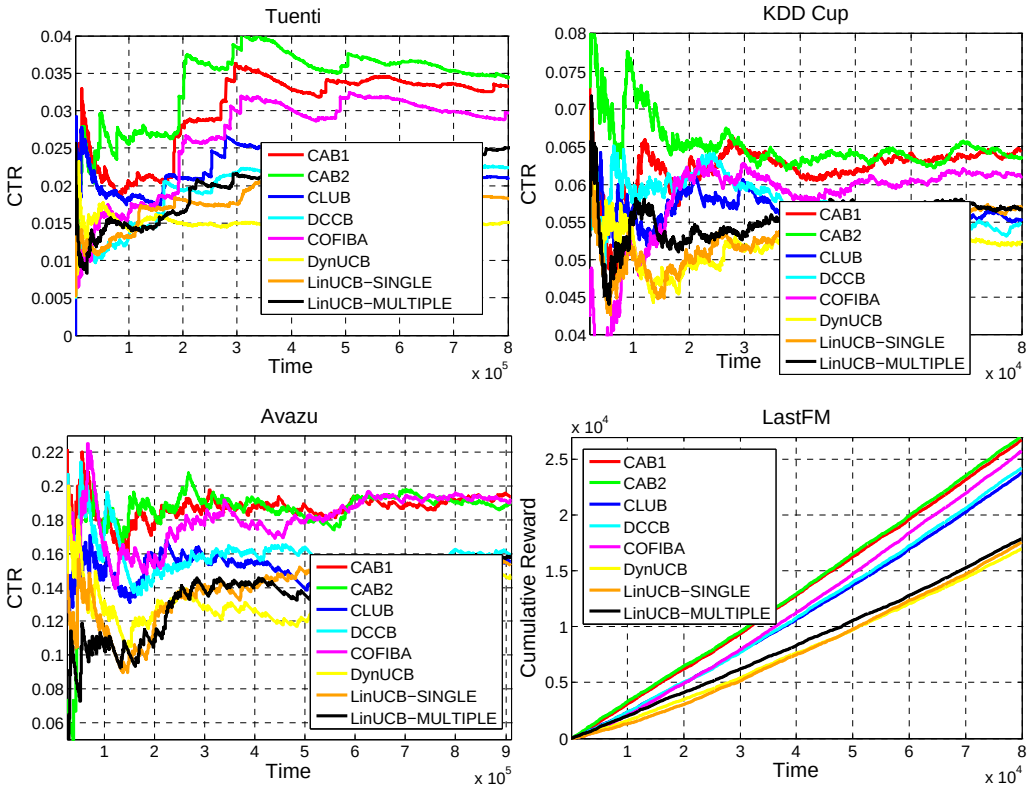

Figure 2: Experimental results on the production and real-world datasets.

- DYNUCB [14] is the traditional "K-Means" algorithm based clustering of bandits;
- LINUCB-SINGLE and LINUCB-MULTIPLE are the single and multiple instances of the LinUCB [12] where it provides the sole and full personalised recommendations respectively.

5.3 Results

The results are summarized in Figures 2. The Tuenti, KDD Cup an Avazu datasets are real online advertising data, thus we measured by the Click-Through Rate (CTR) directly, whereas for the LastFM dataset we measured the the cumulative reward by standard convention therein. These experiments are also aimed at testing the performance of various bandit algorithms in order to provide some insights in terms of prediction accuracy, in particular, with respect to the cold-start regimes in recommender systems.

In all the four datasets, CAB is clearly outperforming the other algorithms. In particular, we observe that in the cold-start period (i.e., the first relative small fraction of time horizon) on all the datasets, CAB tends to perform much better than the alternative methods. We also note that CAB is outperforming the competing methods over the whole time window of the experiments. It's clear that the difference of CAB compared to the other state-of-the-art clustering bandit methods is very significant and typically much bigger than any of the differences among the lower ranked methods. These give further credibility to our claim that it is important to exploit the collaborative effects intrinsically embedded at these data.

The machinery of CAB is scalable and gracefully handles the cold-start problem in recommender systems, where the datasets had large number of users or last for a long time scale. In addition, CAB is exhibiting excellent performance increase and is a principle solution for mitigating the cold-start (a.k.a, the data sparsity issue). CAB achieves surprisingly CTR improvement in the Tuenti dataset which comes from real-world production system, where the CAB almost doubles the CTR

compared to all the other competitors[4], which demonstrates that the idea of exploiting the underlying collaborative effects via CAB is effective.

6 Conclusion

We proposed a bandit method for practical personalised recommendations for Web-based systems. Context-Aware clustering of Bandits (CAB), was derived within the framework of clustering of bandits. Recommendations can be computed sequentially in a computationally efficient and stable manner. CAB exploits context-aware clustering, and thus takes collaborative effects into account, it is scalable and gracefully mitigates the cold-start problem. We provided the theoretical analysis for CAB in a standard stochastic setup. Through experiments on production and real-world datasets, we showed that it achieves significant gains in terms of prediction performance against several state-of-the-art methods.

Acknowledgements

The authors thank Alexandros Karatzoglou for his inputs on an earlier version of the paper.

References

[1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Proc. NIPS*, 2011.

[2] J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

[3] P. Auer. Using confidence bounds for exploration-exploitation trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[4] P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 2001.

[5] Nicolò Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Proc. NIPS*, 2013.

[6] W. Chu, L. Li, L. Reyzin, and R. E Schapire. Contextual bandits with linear payoff functions. In *Proc. AISTATS*, 2011.

[7] K. Crammer and C. Gentile. Multiclass classification with bandit feedback using adaptive regularization. In *Proc. ICML*, 2011.

[8] J. Djolonga, A. Krause, and V. Cevher. High-dimensional gaussian process bandits. In *NIPS*, pages 1025–1033, 2013.

[9] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *Proc. ICML*, 2014.

[10] Nathan Korda, Balazs Szorenyi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *The 33rd International Conference on Machine Learning (ICML)*, 2016.

[11] A. Krause and C.S. Ong. Contextual gaussian process bandit optimization. In *Proc. 25th NIPS*, 2011.

[12] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proc. WWW*, 2010.

[13] Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *The 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2016.

[14] Trong T. Nguyen and Hady W. Lauw. Dynamic clustering of contextual multi-armed bandits. In *Proc. 23rd CIKM*, pages 1959–1962. ACM, 2014.

[4] Notice that in Internet Ads based companies such as Google, Facebook etc, even small CTR enchancement would bring big revenue gains where ad incomes account for more than 95% of total revenues (i.e., 67.39 Billion USD) for Google in 2015

[15] I. Pilaszy and D. Tikk. Recommending new movies: Even a few ratings are more valuable than metadata. In *Proc. RecSys*, 2009.

[16] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 285–295, New York, NY, USA, 2001. ACM.

[17] Y. Seldin, P. Auer, F. Laviolette, J. Shawe-Taylor, and R. Ortner. Pac-bayesian analysis of contextual bandits. In *NIPS*, pages 1683–1691, 2011.

[18] L. Tang, Y. Jiang, L. Li, and T. Li. Ensemble contextual bandits for personalized recommendation. In *Proc. RecSys*, 2014.

[19] Y. Yue, S. A. Hong, and C. Guestrin. Hierarchical exploration for accelerating contextual bandits. In *ICML*, 2012.

A Proof of Lemma 1

Lemma 1. *With probability at least $1-\delta$, uniformly over all pairs of users $i, j \in \mathcal{U}$ and all $t = 1, 2, \ldots$, the following results hold*

1. *If $\left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| > \gamma$ and $\textsc{tcb}_{i,t}(\mathbf{x}) + \textsc{tcb}_{j,t}(\mathbf{x}) \leq \gamma/2$, then $\left|\mathbf{w}_{i,t}^\top \mathbf{x} - \mathbf{w}_{j,t}^\top \mathbf{x}\right| > \textsc{tcb}_{i,t}(\mathbf{x}) + \textsc{tcb}_{j,t}(\mathbf{x})$*
2. *If $\left|\mathbf{w}_{i,t}^\top \mathbf{x} - \mathbf{w}_{j,t}^\top \mathbf{x}\right| > \textsc{tcb}_{i,t}(\mathbf{x}) + \textsc{tcb}_{j,t}(\mathbf{x})$, then $\left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| > \gamma$.*

Proof. The proof of this lemma begins with a basic result on linear bandits [1, Theorem 2] which states that with probability at least $1-\delta$, uniformly over users i, time steps t, and contexts $\mathbf{x}$, we have

$$\left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{w}_{i,t}^\top \mathbf{x}\right| \leq \textsc{tcb}_{i,t}(\mathbf{x}).$$

The above result hinges on the fact that the least-squares regression operation (steps 4, 14, 15 of Algorithm 1) progressively reduces the uncertainty the algorithm has with respect to payoffs for contexts oriented in different directions. Using this result, to prove part 1, we have

$$\begin{aligned}
\gamma &< \left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| \\
&\leq \left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{w}_{i,t}^\top \mathbf{x}\right| + \left|\mathbf{w}_{i,t}^\top \mathbf{x} - \mathbf{w}_{j,t}^\top \mathbf{x}\right| + \left|\mathbf{w}_{j,t}^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| \\
&\leq \textsc{tcb}_{i,t}(\mathbf{x}) + \textsc{tcb}_{j,t}(\mathbf{x}) + \left|\mathbf{w}_{i,t}^\top \mathbf{x} - \mathbf{w}_{j,t}^\top \mathbf{x}\right| \\
&\leq \left|\mathbf{w}_{i,t}^\top \mathbf{x} - \mathbf{w}_{j,t}^\top \mathbf{x}\right| + \gamma/2,
\end{aligned}$$

which establishes the result since we have $\left|\mathbf{w}_{i,t}^\top \mathbf{x} - \mathbf{w}_{j,t}^\top \mathbf{x}\right| > \gamma/2 \geq \textsc{tcb}_{i,t}(\mathbf{x}) + \textsc{tcb}_{j,t}(\mathbf{x})$. To prove part 2, we similarly have

$$\begin{aligned}
\textsc{tcb}_{i,t}(\mathbf{x}) + \textsc{tcb}_{j,t}(\mathbf{x}) &< \left|\mathbf{w}_{i,t}^\top \mathbf{x} - \mathbf{w}_{j,t}^\top \mathbf{x}\right| \\
&\leq \left|\mathbf{w}_{i,t}^\top \mathbf{x} - \mathbf{u}_i^\top \mathbf{x}\right| + \left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| + \left|\mathbf{u}_j^\top \mathbf{x} - \mathbf{w}_{j,t}^\top \mathbf{x}\right| \\
&\leq \left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| + \textsc{tcb}_{i,t}(\mathbf{x}) + \textsc{tcb}_{j,t}(\mathbf{x}),
\end{aligned}$$

which tells us that $\left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| > 0$. However by the context-sensitive well-separatedness assumption, this means that users i and j must lie in different clusters with respect to the item $\mathbf{x}$, and hence $\left|\mathbf{u}_i^\top \mathbf{x} - \mathbf{u}_j^\top \mathbf{x}\right| > \gamma$ which proves the result. □

B Proof of Lemma 3

Lemma 3. *Assume that users and contexts are generated according to the model specified in Section 2. Then, with probability $1-\delta$, the following holds uniformly over all time steps $t = 1, 2, \ldots$, all users $i \in \mathcal{U}$ and all contexts $\mathbf{x} \in \mathbb{R}^d$ of less than unit norm,*

$$\textsc{tcb}_{i,t}(\mathbf{x}) \leq U_{d,\lambda,\mu,\delta}(i,t) := \frac{3\sigma\sqrt{d\log t + \log(1/\delta)}}{\sqrt{1 + B_{\lambda,\nu}(T_{i,t}, \delta/2nd)}},$$

where $B_{\lambda,\nu}(T,\delta) = (C(\lambda,\nu)\cdot T - 8(\log(T/\delta) + \sqrt{T\log(T/\delta)}))_+$ *and* $C(\lambda,\nu)$ *is a strictly positive constant that depends on* λ*, the smallest eigenvalue of the covariance matrix associated with the distribution* $\mathcal{D}$ *and* ν^2*, its conditional sub-Gaussian variance parameter.*

Proof. The proof of this lemma is similar to that of [9, Lemma 2], with the main difference being in bounding the expected value of the quantity $\mathbb{E}_t\left[\min_{k=1,2,\ldots,c_t}(\mathbf{z}^\top \mathbf{x}_{t,i})^2 \middle| (i_t, c_t)\right]$ for any fixed vector $\mathbf{z} \in \mathbb{R}^d$. Whereas [9] make (strong) additional assumptions about the sub-Gaussian norm of the distribution $\mathcal{D}$ to prove their result [9, Lemma 2], we show here that an unconditional bound is possible as well. We begin by noticing that using standard tail bounds for sub-Gaussian variables, we have, for any fixed $k \le c_t$ and $\mathbf{z} \in \mathbb{R}^d$,

$$\mathbb{P}_t\left[(\mathbf{z}^\top \mathbf{x}_{t,k})^2 > \lambda - a \middle| (i_t, c_t)\right] \ge (1 - 2e^{-a^2/2\nu^2}),$$

where $\mathbb{P}_t[\cdot] := \mathbb{P}\left[\cdot | (i_1, C_{i_1}, a_1), (i_2, C_{i_2}, a_2), \ldots, (i_{t-1}, C_{i_{t-1}}, a_{t-1})\right]$. This gives us

$$\mathbb{P}_t\left[\min_{k=1,2,\ldots,c_t}(\mathbf{z}^\top \mathbf{x}_{t,k})^2 > \lambda - a \middle| (i_t, c_t)\right] \ge (1 - 2e^{-a^2/2\nu^2})^{c_t},$$

since the context are (conditionally) chosen independently. Now, since $\min_{k=1,2,\ldots,c_t}(\mathbf{z}^\top \mathbf{x}_{t,k})^2$ is a positive valued random variable, we can bound its expectation using its complimentary cumulative distribution function, which we have already bounded above. Let K be an upper bound on the number of items being recommended at any time step, i.e. $c_t \le K$ for all t. Then we have

$$\begin{aligned}
\mathbb{E}_t\left[\min_{k=1,2,\ldots,c_t}(\mathbf{z}^\top \mathbf{x}_{t,i})^2 \middle| (i_t, c_t)\right] &= \int_0^\infty \mathbb{P}_t\left[\min_{k=1,2,\ldots,c_t}(\mathbf{z}^\top \mathbf{x}_{t,k})^2 > \zeta \middle| (i_t, c_t)\right] d\zeta \\
&\ge \int_0^\lambda \mathbb{P}_t\left[\min_{k=1,2,\ldots,c_t}(\mathbf{z}^\top \mathbf{x}_{t,k})^2 > \zeta \middle| (i_t, c_t)\right] d\zeta \\
&\ge \int_0^\lambda (1 - 2e^{-(\lambda-\zeta)^2/2\nu^2})^{c_t}\, d\zeta \\
&\ge \int_0^\lambda (1 - 2e^{-(\lambda-\zeta)^2/2\nu^2})^{K}\, d\zeta \\
&=: C(\lambda,\nu).
\end{aligned}$$

Proceeding further as in [9, Lemma 2] by setting up a Freedman-style concentration bound to get a high-confidence estimate then finishes the proof. □

C Proof of Lemma 6

Lemma 6. *With probability at least* $1-\delta$*, we have, for both the* CAB*1 and the* CAB*2 updates*

$$(B) \le 2n\left(\max\left\{\frac{32^3}{C(\lambda,\nu)^3}\log^2\frac{1}{\delta}, \frac{288\sigma^2\left(d\log T + \log(1/\delta)\right)}{C(\lambda,\nu)\cdot\gamma^2}\right\} + 2\log\frac{nT}{\delta}\right).$$

Proof. We begin by noticing that if $T_{i,t} \ge T_0 := \max\left\{\frac{32^3}{C(\lambda,\nu)^3}\log^2\frac{1}{\delta}, \frac{288\sigma^2(d\log T+\log(1/\delta))}{C(\lambda,\nu)\cdot\gamma^2}\right\}$, then we have $M_{d,\lambda,\mu,\delta}(i,t) \le \gamma/4$. This gives us

$$(B) = \sum_{t=1}^T \mathbb{I}\left\{\max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t) > \gamma/4\right\} \le \sum_{t=1}^T \mathbb{I}\left\{\min_{i\in\mathcal{U}} T_{i,t} < T_0\right\}.$$

Now, since $T_{i,t} = \sum_{\tau=1}^t \mathbb{I}\{i_\tau = i\}$ and each of the random variables $\mathbb{I}\{i_\tau = i\}$ is a Bernoulli random variable with parameter $\frac{1}{n}$, we have $\mathbb{E}[T_{i,t}] = \frac{t}{n}$. Applying the Bernstein inequality for Bernoulli variables and taking a union bound over all time steps, and all n users $i \in \mathcal{U}$ (note that $|\mathcal{U}| = n$) tells us that

$$\mathbb{P}\left[\exists i \in \mathcal{U}, \exists t > 2n\left(T_0 + 2\log\frac{nT}{\delta}\right) : T_{i,t} < T_0\right] \le \delta,$$

which tells us that with probability at least $1-\delta$, we have

$$\sum_{t=1}^T \mathbb{I}\left\{\min_{i\in\mathcal{U}} T_{i,t} < T_0\right\} \le 2n\left(T_0 + 2\log\frac{nT}{\delta}\right)$$ □

D Proof of Lemma 7

Lemma 7. *With probability at least $1-\delta$, we have, for the* CAB*1 update*

$$(A) \leq 3\sigma\sqrt{d\log T+\log(1/\delta)}\cdot\left(2n\left(\frac{32^3}{C(\lambda,\nu)^3}\log^2\frac{1}{\delta}+2\log\frac{nT}{\delta}\right)+4n\log\frac{nT}{\delta}+\frac{8n}{C(\lambda,\nu)}+4\sqrt{\frac{nT}{C(\lambda,\nu)}}\right)$$

Proof. We first note that at any time step t, if $\max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t)\leq\gamma/4$, then by applying Corollary 2 and Lemma 3, we can conclude that for all items $\mathbf{x}_{t,k}\in C_{i_t}$, we have $N_{t,k}=U_{\mathfrak{c}(i_t,\mathbf{x}_{t,k})}(\mathbf{x}_{t,k})$, i.e. at these time steps, the CAB algorithm correctly identifies the neighborhood of the user i_t with respect to all items under consideration. This allows us to write

$$\begin{aligned}
&r_t\cdot\mathbb{I}\left\{\max_{i\in\mathcal{U}}M_{d,\lambda,\mu,\delta}(i,t)\leq\gamma/4\right\}=\mathbf{u}_{i_t}^\top(\mathbf{x}_t^*-\widehat{\mathbf{x}}_t)\cdot\mathbb{I}\left\{\max_{i\in\mathcal{U}}M_{d,\lambda,\mu,\delta}(i,t)\leq\gamma/4\right\}\\
&=\left(\frac{1}{|N_t^*|}\sum_{j\in N_t^*}\mathbf{u}_j^\top\mathbf{x}_t^*-\frac{1}{\left|\widetilde{N}_t\right|}\sum_{j\in\widetilde{N}_t}\mathbf{u}_j^\top\widehat{\mathbf{x}}_t\right)\cdot\mathbb{I}\left\{\max_{i\in\mathcal{U}}M_{d,\lambda,\mu,\delta}(i,t)\leq\gamma/4\right\}\\
&=\frac{1}{\left|\widehat{N}_t^*\right|}\sum_{j\in\widehat{N}_t^*}\mathbf{u}_j^\top\mathbf{x}_t^*-\frac{1}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}\mathbf{u}_j^\top\widehat{\mathbf{x}}_t\\
&\leq\frac{1}{\left|\widehat{N}_t^*\right|}\sum_{j\in\widehat{N}_t^*}(\mathbf{w}_{j,t-1}^\top\mathbf{x}_t^*+\textsc{tcb}_{j,t-1}(\mathbf{x}_t^*))-\frac{1}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}(\mathbf{w}_{j,t-1}^\top\widehat{\mathbf{x}}_t-\textsc{tcb}_{j,t-1}(\widehat{\mathbf{x}}_t))\\
&\leq\frac{1}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}(\mathbf{w}_{j,t-1}^\top\widehat{\mathbf{x}}_t+\textsc{tcb}_{j,t-1}(\widehat{\mathbf{x}}_t))-\frac{1}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}(\mathbf{w}_{j,t-1}^\top\widehat{\mathbf{x}}_t-\textsc{tcb}_{j,t-1}(\widehat{\mathbf{x}}_t))\\
&=\frac{2}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}\textsc{tcb}_{j,t-1}(\widehat{\mathbf{x}}_t).
\end{aligned}$$

In the above sequence, the second step uses the context-sensitive well-separatedness assumption. The third step uses the fact that all neighborhoods were identified exactly. The fourth step uses the result from [1, Theorem 2] which states that with probability at least $1-\delta$, uniformly over users i, time steps t, and contexts $\mathbf{x}$, we have $\left|\mathbf{u}_i^\top\mathbf{x}-\mathbf{w}_{i,t}^\top\mathbf{x}\right|\leq\textsc{tcb}_{i,t}(\mathbf{x})$. The fifth step follows from step 12 in Algorithm 1 which selected the neighborhood $\widehat{N}_t$ over $\widehat{N}_t^*$ (recall that for the sake of simplicity, we assumed that Algorithm 1 is executed with the TCB expressions providing the confidence bounds instead of the CB expressions).

Now, using Lemma 3, we get,

$$\frac{1}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}\textsc{tcb}_{j,t-1}(\widehat{\mathbf{x}}_t)\leq\frac{1}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}M_{d,\lambda,\mu,\delta}(j,t-1)\leq\max_{i\in\mathcal{U}}M_{d,\lambda,\mu,\delta}(i,t-1)$$

Thus, we get

$$(A)=\sum_{t=1}^{T}r_t\cdot\mathbb{I}\left\{\max_{i\in\mathcal{U}}M_{d,\lambda,\mu,\delta}(i,t)\leq\gamma/4\right\}\leq\sum_{t=1}^{T-1}\max_{i\in\mathcal{U}}M_{d,\lambda,\mu,\delta}(i,t).$$

Now, using slighly tedious calculations, we can show that

$$M_{d,\lambda,\mu,\delta}(i,t)\leq3\sigma\sqrt{d\log T+\log(1/\delta)}\left(\mathbb{I}\{T_{i,t}<T_1\}+\mathbb{I}\{T_{i,t}\geq T_0\}\cdot\frac{1}{\sqrt{1+C(\lambda,\nu)/2\cdot T_{i,t}}}\right),$$

where $T_1=\frac{32^3}{C(\lambda,\nu)^3}\log^2\frac{1}{\delta}$. Now using Bernstein bounds for Bernoulli variables as before, we have, with probability at least $1-\delta$,

$$\sum_{t=1}^{T}\max_{i\in\mathcal{U}}\mathbb{I}\{T_{i,t}<T_1\}\leq2n\left(T_1+2\log\frac{nT}{\delta}\right).$$

Before we move on to bound the other part, we notice that using Bernstein bounds, we can also get the following result

$$\mathbb{P}\left[\exists t > 0, \exists i \in \mathcal{U} : T_{i,t} \leq \frac{t}{2n} - 2\log\frac{nT}{\delta}\right] \leq \delta,$$

which lets us bound the other quantity as follows

$$\begin{aligned}\sum_{t=1}^{T} \max_{i\in\mathcal{U}} \left\{ \frac{\mathbb{I}\{T_{i,t} \geq T_0\}}{\sqrt{1 + C(\lambda,\nu)/2 \cdot T_{i,t}}} \right\} &\leq \sum_{t=1}^{T} \frac{1}{\sqrt{1 + \frac{C(\lambda,\nu)}{2} \cdot \left(\frac{t}{2n} - 2\log\frac{nT}{\delta}\right)_+}} \\ &\leq 4n\log\frac{nT}{\delta} + \frac{8n}{C(\lambda,\nu)} + 4\sqrt{\frac{nT}{C(\lambda,\nu)}}.\end{aligned}$$

Combining the two estimates concludes the proof. □

E Proof of Theorem 9

Theorem 9. *With probability at least $1-\delta$, uniformly over time steps $t > 0$, users $i \in \mathcal{U}$, and context vectors $\mathbf{x}_{t,k} \in C_{i_t}$, we have*

$$\left| \left(\overline{\mathbf{w}}_{t,k} - \frac{1}{|N|} \sum_{j\in N} \mathbf{u}_j \right)^\top \mathbf{x}_{t,k} \right| \leq \overline{\mathrm{TCB}}_{t,k}(\mathbf{x}_{t,k}) + \mathcal{O}\left(\kappa_{i_t,\mathbf{x}_{t,k}}\right),$$

where $N = \mathfrak{c}(i_t, \mathbf{x})$ denotes the true neighborhood of the user i_t with respect to an item $\mathbf{x}_{t,k} \in C_{i_t}$ and $\kappa_{i_t,\mathbf{x}_{t,k}}$ denotes the discrepancy in the neighborhood of the user i_t with respect to the item $\mathbf{x}_{t,k}$.

Proof. For every user $j \in N$, let $X_j \in \mathbb{R}^{d\times T_{i,t-1}}$ be the matrix containing the context vector for time instances in the past when user j was served. Also let $\mathbf{a}_j \in \mathbb{R}^{T_{i,t-1}}$ denote the vector of payoffs the user j gave in his previous time instances, and $\boldsymbol{\epsilon}_j \in \mathbb{R}^{T_{i,t-1}}$ denote the vector of noise values user j infused in his payoffs in those time instances.

Then we have (as described in Algorithm 3, the CAB2 update)

$$\begin{aligned}\overline{\mathbf{w}}_{t,k} &= \left(\sum_{j\in N} X_j X_j^\top + I\right)^{-1} \left(\sum_{j\in N} X_j \mathbf{a}_j\right) \\ &= \left(\sum_{j\in N} X_j X_j^\top + I\right)^{-1} \left(\sum_{j\in N} X_j \left(X_j^\top \mathbf{u}_j + \boldsymbol{\epsilon}_j\right)\right).\end{aligned}$$

Now notice that since all these users belong to the same cluster with respect to the context $\mathbf{x}_{t,k}$, the context-sensitive well separated condition dictates that their model vectors $\mathbf{u}_j$ have the same projection onto the context $\mathbf{x}_{t,k}$. Thus, we decompose $\mathbf{u}_j = \mathbf{v}^{\|} + \mathbf{v}_j^{\perp}$ where the vector $\mathbf{v}^{\|}$ is parallel to the context vector $\mathbf{x}_{t,k}$ and the vectors $\mathbf{v}_j^{\perp}$ are all perpendicular to $\mathbf{x}_{t,k}$.

Note that we make a crucial observation here that all the vectors $\mathbf{u}_j$ share the component parallel to the context vector $\mathbf{x}_{t,k}$. This happens because all of them have the same inner product with that context vector. We will denote, as a shorthand $X = [X_{j_1} X_{j_2} \dots X_{j_{|N|}}]$, $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_{j_1}^\top \boldsymbol{\epsilon}_{j_2}^\top \dots \boldsymbol{\epsilon}_{j_{|N|}}^\top]^\top$. For any $j_k \in N$ $X_{\setminus j_k}$ will be the matrix with X_j concatenated but the matrix j_k left out. This gives us

$$\begin{aligned}\overline{\mathbf{w}}_{t,k} &= (XX^\top + I)^{-1} \sum_{j\in N} X_j \left(X_j^\top (\mathbf{v}^{\|} + \mathbf{v}_j^{\perp}) + \boldsymbol{\epsilon}_j\right) \\ &= \mathbf{v}^{\|} + \left(XX^\top + I\right)^{-1} \mathbf{v}^{\|} + (XX^\top + I)^{-1} \sum_{j\in N} X_j X_j^\top \mathbf{v}_j^{\perp} + (XX + I)^{-1} X\boldsymbol{\epsilon}\end{aligned}$$

$$= \underbrace{\mathbf{v}^{\|}}_{(P)} + \underbrace{\left(XX^\top + I\right)^{-1}\mathbf{v}^{\|}}_{(Q)} + \underbrace{(XX^\top + I)^{-1}\sum_{j\in N} X_j X_j^\top \mathbf{v}_j^{\perp}}_{(R)} + \underbrace{(XX + I)^{-1} X\boldsymbol{\epsilon}}_{(S)}.$$

Analyzing (P) By construction, $\mathbf{x}_{t,k}^\top \mathbf{v}^{\|}$ is the expected payoff every user $j \in N$ gives the context $\mathbf{x}_{t,k}$. Thus, $\mathbf{x}_{t,k}^\top \mathbf{v}^{\|} = \frac{1}{|N|}\sum_{j\in N} \mathbf{u}_j^\top \mathbf{x}_{t,k}$.

Analyzing (Q) As all model vectors are at most unit norm, we can see that $\mathbf{x}_{t,k}^\top \left(XX^\top + I\right)^{-1}\mathbf{v}^{\|} = \left\langle \mathbf{x}_{t,k}, \mathbf{v}^{\|}\right\rangle_{\overline{M}_{t,k}^{-1}} \le \|\mathbf{x}_{t,k}\|_{\overline{M}_{t,k}^{-1}}$.

Analyzing (S) Using the *Self-Normalized Bound for Vector-Valued Martingales* from [1, Theorem 1], w.p. $1-\delta$, $\mathbf{x}_{t,k}^\top (XX+I)^{-1}(X\boldsymbol{\epsilon}) \le \|\mathbf{x}_{t,k}\|_{\overline{M}_{t,k}^{-1}} \|X\boldsymbol{\epsilon}\|_{\overline{M}_{t,k}^{-1}} \le \|\mathbf{x}_{t,k}\|_{\overline{M}_{t,k}^{-1}} \sigma\sqrt{2\log\frac{2|\overline{M}_{t,k}|}{\delta}}$.

Analyzing (R) Let $\Delta_j = \mathbf{v}_j^\perp - \mathbf{v}_1^\perp$. Then we have

$$\begin{aligned}(XX^\top + I)^{-1}\sum_{j\in N}(X_jX_j^\top)\mathbf{v}_j^\perp &= (XX^\top+I)^{-1}(XX^\top)\mathbf{v}_1^\perp + (XX^\top+I)^{-1}\sum_{j\in N}(X_jX_j^\top)\Delta_j \\ &= \mathbf{v}_1^\perp + (XX^\top+I)^{-1}\mathbf{v}_1^\perp + (XX^\top+I)^{-1}\sum_{j\in N}(X_jX_j^\top)\Delta_j.\end{aligned}$$

Now, by construction, we have $\mathbf{x}_{t,k}^\top \mathbf{v}_1^\perp = 0$. Moreover, since all model vectors are at most unit norm, we have $\mathbf{x}_{t,k}^\top (XX^\top+I)^{-1}\mathbf{v}_1^\perp \le \|\mathbf{x}_{t,k}\|_{\overline{M}_{t,k}^{-1}}$. For the last term, we resort to a brute force bound using the fact that $\|\Delta_j\|_2 = \|\mathbf{u}_j - \mathbf{u}_1\|_2 \le \kappa_{i_t,\mathbf{x}_{t,k}}$.

Since all context vectors are at most unit norm, we have $\|\mathbf{x}_{t,k}\|_2 \le 1$. For the same reason,

$$\left\|(X_jX_j^\top)\Delta_j\right\|_2 \le T_{j,t-1}\cdot\kappa_{i_t,\mathbf{x}_{t,k}}$$

Moreover, Using results on the singular values of $X_jX_j^\top$ (see [9] Lemma 2 and proof thereof), we have $\left\|(XX^\top+I)^{-1}\right\| \le \mathcal{O}\left(\frac{1}{C(\lambda,\nu)\cdot T_{N,t-1}}\right)$. Since $T_{N,t-1} = \sum_{j\in N} T_{j,t-1}$, applying the Cauchy-Schwartz inqeuality to bound the quantity $\mathbf{x}_{t,k}^\top\left((XX^\top+I)^{-1}\sum_{j\in N}(X_jX_j^\top)\Delta_j\right)$ and putting all the bounds together establishes the claimed result. □

F Proof of Lemma 10

The proof begins in a manner similar to that in the analysis of CAB1 but later diverges. We first restate the lemma and then proceed with the proof

Lemma 10. *With probability at least $1-\delta$, for the* CAB2 *update with uniform neighborhoods,*

$$(A) \le 6(\sigma\sqrt{d\log T + \log(1/\delta)}+n)\cdot\left(2n\left(\frac{(32n)^3}{C(\lambda,\nu)^3}\log^2\frac{1}{\delta} + 2n\log\frac{T}{\delta}\right) + 4n^2\log\frac{T}{\delta} + \frac{8n}{C(\lambda,\nu)} + 4\sqrt{\frac{nT}{C(\lambda,\nu)}}\right)$$

Going as before, we have

$$\begin{aligned} r_t\cdot\mathbb{I}\left\{\max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t)\le\gamma/4\right\} &= \mathbf{u}_{i_t}^\top(\mathbf{x}_t^* - \widehat{\mathbf{x}}_t)\cdot\mathbb{I}\left\{\max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t)\le\gamma/4\right\} \\ &= \left(\frac{1}{|N_t^*|}\sum_{j\in N_t^*}\mathbf{u}_j^\top\mathbf{x}_t^* - \frac{1}{\left|\widetilde{N}_t\right|}\sum_{j\in\widetilde{N}_t}\mathbf{u}_j^\top\widehat{\mathbf{x}}_t\right)\cdot\mathbb{I}\left\{\max_{i\in\mathcal{U}} M_{d,\lambda,\mu,\delta}(i,t)\le\gamma/4\right\} \\ &= \frac{1}{\left|\widehat{N}_t^*\right|}\sum_{j\in\widehat{N}_t^*}\mathbf{u}_j^\top\mathbf{x}_t^* - \frac{1}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}\mathbf{u}_j^\top\widehat{\mathbf{x}}_t.\end{aligned}$$

At this point the analysis diverges from the one done for the CAB1 aggregation technique since the model vector used to estimate the neighborhood level payoff for a neighborhood $N_{t,k}$ is not simply $\frac{1}{|N_{t,k}|}\sum_{j\in N_{t,k}} \mathbf{w}_{j,t-1}$ (as it was for the CAB1 approach) but a re-estimation of the least-squares model vector for the combined histories of all users in that neighborhood.

To remedy this, we use Theorem 9 to get

$$\begin{aligned}
\frac{1}{\left|\widehat{N}_t^*\right|}\sum_{j\in\widehat{N}_t^*}\mathbf{u}_j^\top\mathbf{x}_t^* - \frac{1}{\left|\widehat{N}_t\right|}\sum_{j\in\widehat{N}_t}\mathbf{u}_j^\top\widehat{\mathbf{x}}_t &\leq (\overline{\mathbf{w}}_{t,k_t^*}^\top\mathbf{x}_t^* + \overline{\text{TCB}}_{t,k}(\mathbf{x}_t^*)) - (\overline{\mathbf{w}}_{t,k_t}^\top\widehat{\mathbf{x}}_t - \overline{\text{TCB}}_{t,k}(\widehat{\mathbf{x}}_t)) \\
&\leq (\overline{\mathbf{w}}_{t,k_t}^\top\widehat{\mathbf{x}}_t + \overline{\text{TCB}}_{t,k}(\widehat{\mathbf{x}}_t)) - (\overline{\mathbf{w}}_{t,k_t}^\top\widehat{\mathbf{x}}_t - \overline{\text{TCB}}_{t,k}(\widehat{\mathbf{x}}_t)) \\
&= 2\overline{\text{TCB}}_{t,k}(\widehat{\mathbf{x}}_t) \\
&\leq 2\overline{M}_{d,\lambda,\mu,\delta}(t,k_t),
\end{aligned}$$

where the second step follows from step 12 in Algorithm 1 which selected $\widehat{N}_t$ instead of $\widehat{N}_t^*$ and the last step follows from an application of Lemma 4. Now since $T_{N_{t,k_t},t} \geq T_{i,t}$ as the total number of times neighbors of i_t have been served include the times i_t itself has been served, similar arguments can be made as done for the CAB1 analysis to finish the proof.